

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A method comprising:

initializing a computer system, the computer system including ~~a first processor an~~
application processor to execute applications and processes in the computer system and a
~~second processor a bootstrap processor to initialize the computer system;~~

~~designating the first processor to handle~~ declaring the application processor to be
dedicated to handling a polling function for a timer interrupt process for the computer
system, wherein a normal execution thread is to be processed by the ~~second processor~~
bootstrap processor, and wherein the timer interrupt process is the only method of
asynchronous event handling that is available to the computer system;

setting a timer for a plurality of time intervals for the timer interrupt process;

calling a polling function at the end of each of the plurality of time intervals, the
polling function being performed by the ~~first processor~~ application processor, the polling
function to determine if any special events have occurred; and

if the polling function results in a positive result, processing the results of the
polling function with the ~~second processor~~ bootstrap processor.

2. (Currently amended) The method of claim 1, wherein the polling function
performed by the application processor comprises event handling for a network stack by
polling a network interface card (NIC) of the computer system.

3. (Currently amended) The method of claim 1, wherein ~~the first processor is an application processor for the computer system~~ wherein the polling function provides for communications between a network protocol layer and a network interface layer for the computer system.
4. (Currently amended) The method of ~~claim 3~~ claim 1, further comprising ~~declaring the first processor to be dedicated to the polling function;~~ wherein a special event regards data that is available to be transferred from a network stack.
5. (Cancelled)
6. (Currently amended) The method of claim 1, wherein the normal execution thread is processed by the ~~second processor~~ bootstrap processor in parallel at least in part with performance of the polling function by the ~~first processor~~ application processor.
7. (Cancelled)
8. (Currently amended) An event handling mechanism for a computer system comprising:
a first processor application processor, the first processor designated application processor being dedicated to handle a timer interrupt process, the first processor to perform a polling operation for event handling each time an interrupt timer reaches a specified time interval, the timer interrupt process being the only method of asynchronous event handling available for the computer system; and
a second processor bootstrap processor, the second processor bootstrap processor to perform a normal processing operation, the ~~first processor~~ application processor to

transfer data to the ~~second-processor bootstrap processor~~ for processing if the polling operation ~~provides a positive result~~ detects an event.

9. (Currently amended) The event handling mechanism of claim 8, wherein the performance of the polling operation by the ~~first-processor~~ application processor overlaps at least in part with the performance of the normal processing operation by the ~~second-processor bootstrap processor~~.

10. (Currently amended) The event handling mechanism of claim 8, wherein ~~the first processor is dedicated to event handling~~ an event regards data that is available to be transferred from a network stack.

11. (Currently amended) The event handling mechanism of claim 8, wherein an event for the polling operation for the computer system comprises a network stack event and wherein the ~~first-processor~~ application processor polls a network interface card (NIC) of the computer system.

12. (Currently amended) The event handling mechanism of claim 8, wherein the ~~first processor~~ application processor and the ~~second-processor bootstrap processor~~ are separate physical processors of the computer system.

13. (Currently amended) The event handling mechanism of claim 8, wherein the ~~first processor~~ application processor and the ~~second-processor bootstrap processor~~ are logical processors in a single physical processor of the computer system.

14. (Currently amended) A computer system comprising:

~~a first-processor~~ an application processor, the ~~first-processor~~ application processor being ~~designated to perform~~ dedicated to performing an event handling function for the computer system;

~~a second-processor~~ bootstrap processor, the ~~second-processor~~ bootstrap processor to perform a processing function for the computer system;

a timer, the timer being set for a time interval for the event handling function of the ~~first-processor~~ application processor, a function call for the ~~first-processor~~ application processor being called at the end of the time interval for polling of events, the timer providing the only event handling mechanism available for the computer system; and

a network interface, the event handling function providing for monitoring of the network interface for data that is available for transfer; and

a memory, the ~~first-processor~~ application processor writing the data relating to events that is available for transfer to the memory ~~to transfer the data to the second processor to provide such data to the bootstrap processor for processing.~~

15. (Currently amended) The computer system of claim 14, wherein ~~the second processor is a bootstrap processor~~ the data that is available for transfer is transferred from a network stack.

16. (Cancelled)

17. (Currently amended) The computer system of claim 14, wherein the ~~first processor~~ application processor and the ~~second-processor~~ bootstrap processor operate in parallel at least in part.

18. (Cancelled)
19. (Original) The computer system of claim 14, wherein the computer system comprises a single-threaded processing environment.
20. (Currently amended) The computer system of claim 14, wherein the computer system is a multi-processor system, and wherein the ~~first processor~~ application processor is a first physical processor and the ~~second processor~~ bootstrap processor is a second physical processor.
21. (Currently amended) The computer system of claim 14, wherein the computer system is a hyper-threaded system, and wherein the ~~first processor~~ application processor is a first logical processor of a physical processor and the ~~second processor~~ bootstrap processor is a second logical processor of the physical processor.
22. (Currently amended) A ~~machine-readable~~ computer-readable medium having stored thereon data representing sequences of instructions that, when executed by a processor, cause the processor to perform operations comprising:
- initializing a computer system, the computer system including a ~~first processor~~ an application processor to execute applications and processes in the computer system and a ~~second processor~~ bootstrap processor to initialize the computer system;
- designating the ~~first processor~~ application processor as being dedicated to handle a polling function for a timer interrupt process for the computer system, wherein a normal execution thread is to be processed by the second processor wherein the timer interrupt process is the only method of asynchronous event handling that is available to the

computer system, wherein the polling function provides for polling of a network interface for data packets that are ready for transfer;

setting a timer for a plurality of time intervals for the timer interrupt process;

calling a polling function at the end of each of the plurality of time intervals, the polling function being directed to the ~~first processor~~ application processor, the polling function to determine if ~~any special events have occurred~~ data packets are ready for transfer; and

if the polling function results in a positive result, ~~directing the results of the polling function~~ transferring the data packets to the ~~second processor~~ bootstrap processor.

23-26. (Cancelled)

27. (Currently amended) The medium of claim 22, wherein processing of ~~[[a]]~~ the normal execution thread by the ~~second processor~~ bootstrap processor overlaps in time at least in part with performance of the polling function by the ~~first processor~~ application processor.